International Journal of Exploring Emerging Trends in Engineering (IJEETE)
Vol. 03, Issue 02, Mar-Apr, 2016 Pg. 57-66 WWW.IJEETE.COM

GESTURE RECOGNITION USING IMAGE PROCESSING AND CONVERSION TO TEXT AND SPEECH

Kajal B. Borole
Department of Computer Engineering,
Government College of Engineering, Jalgaon
kajalborole7@gmail.com

Bhagyashree D. Patil
Department of Computer Engineering,
Government College of Engineering, Jalgaon
pbhagyashree39@gmail.com

H. D. Gadade
Department of Computer Engineering,
Government College of Engineering ,Jalgaon
gadade4u@gmail.com

Abstract- Gesture recognition is the technique which is used for the interaction between human and computer or it can be for any automation implementation purpose. In this paper, we have made use of the implemented concepts of basic gesture recognition system and further we propose a new attachment to this concept i.e. - to convert these gestures to text and speech. Here we are comparing hand gestures with the specified vocabulary of predefined gestures to recognize the gesture. The paper covers all the necessary idea and algorithms which help in converting gestures to text and Speech. The main aim of this paper is to study and understand the different Algorithms and tools used in the entire paper, i.e.- it includes some implemented algorithms for gesture recognition as a revision and also the use of proposed algorithms for converting the gestures to text and speech which are implemented with help of many API of different programming languages can be used.

Keywords- *CAMshift algorithm,Background Substraction algorithm, Gesture Detection, Text To Speech Conversion*

I. INTRODUCTION

Human hand gestures provide the important and effective means of non-verbal communication with the human and computer. Hand Gestures are meaningful expressive body motions that are movements of hands, fingers arms. Hand gestures ranges from simple identical gestures or static gestures that are used to convert these gestures to voice or text.

Gesture recognition is a process of recognizing the gestures made by the user by any computer or machine. This also gives a human computer interaction. This kind of technique provides in natural way to interact with the machine in a natural way.

The working of the paper is as follows:

The essential aim of building hand gesture recognition system is to create a natural interaction between human and computer where the recognized gestures can be used for controlling a robot or conveying meaningful information [2]. How to form the resulted hand gestures to be understood and well interpreted by the computer considered as the problem of gesture interaction [2].

Recognition of human gestures is an active area of research integral for the development of intuitive human–machine interfaces for ubiquitous

computing and assistive robotics. In particular, such systems are key to effective environmental designs that facilitate aging in place. Typically, gesture recognition takes the form of template matching in which the human participant is expected to emulate a choreographed motion as prescribed by the researchers.

II. RELATED WORK

This section of paper describes the related work and concepts used for conversion of gestures to text and speech. Also a brief revision about the concepts used for gesture recognition.

A] Brief revision of concepts used for gesture recognition:

1.OpenCV (Open Source Computer Vision Library: http://opencv.org) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms. The document describes the so-called OpenCV 2.x API, which is essentially a C++ API, as opposite to the C-based OpenCV 1.x API. The latter is described in opencv1x.pdf.OpenCV has a modular structure, which means that the package includes several shared or static libraries.[4]

2. Basic Operation in Image Processing:

Basic operations in image processing are shown in fig1.

Image acquisition is the first process in which gave some hints regarding the origin of digital images.

Image enhancement is the process of manipulating an image so that the result is more suitable than the original for a specific application. The word *specific* is important here, because it establishes at the outset that enhancement techniques are problem oriented. Thus, for example, a method that is quite useful for enhancing.[20]

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation. Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a "good" enhancement result.

Color image processing is an area that has been gaining in importance because of the significant increase in the use of digital images over the Internet.[20]

Wavelets are the foundation for representing images in various degrees of resolution. In particular, this material is used in this book for image data compression and for pyramidal representation, in which images are subdivided successively into smaller regions.[7]

Compression, as the name implies, deals with techniques for reducing the storage required saving an image, or the bandwidth required transmitting it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar (perhaps inadvertently) to most users of computers in the form of image file extensions, such as the jpg file extension used in the PEG (Joint Photographic Experts Group) image compression standard.

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. The material in this chapter begins a transition from processes that output images to processes that output image attributes.[20]

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually. On the other hand, weak or erratic segmentation

algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition into succeeds. [20]

Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary oaf region (i.e., the set of pixels separating one image region from another) or althea points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. In some applications, these representations complement each other. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. [20]

Description, also called *feature selection*, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.[20]

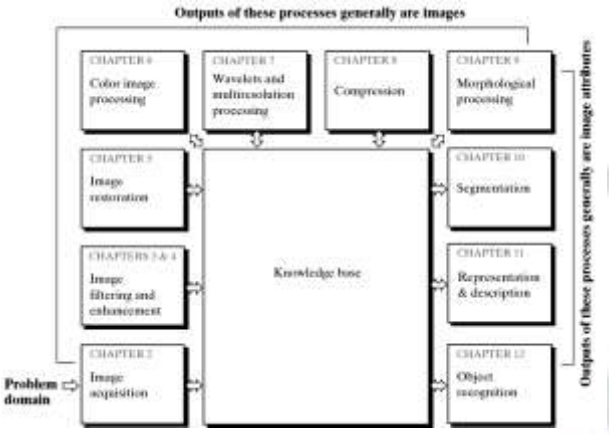

Fig1. Block diagram of basic operation in image processing [20]

III. CONCEPTS AND ALGORITHMS

Now, let's see this entire process of gesture recognition and conversion to text and speech in following sequence:

1. Image Capturing

In this paper, we proposed to capture a gesture using webcam available on our device ex. Laptop. After capturing a gesture, further processes are applied on it.

2. Thresholding

Thresholding enables to achieve image segmentation in the easiest way. Image segmentation means dividing the complete image into a set of pixels in such a way that the pixels in each set have some common characteristics. Image segmentation is highly useful in defining objects and their boundaries. In this chapter we perform some basic thresholding operations on images. We use function threshold. It can be found under Img proc package.[16]

HSV THRESHOLD:

The class HSVThreshold is a tool for clustering problems in HSV (Hue, Saturation, and Value) color space for color-based pixel separation. The main purpose of this tool is to support semi-automated color-based segmentation. Particularly, this tool provides users the decision support about valid pixel value ranges for specific types of object detection.The class operates in three ranges of HSV values. Points within the user-specified value ranges are appeared in the result image, both as RGB and HSV images. In each scrollbar, a user can adjust the upper and lower limits of the pixel values, and can fix the dynamic value range by checking the "Fix Range" option. For example, the lower/upper limit will move automatically while keeping the same dynamic value range when changing the upper/lower limit.

Simple Algorithm

1.Start set THRESHOLD = 30
hue = getHuePixelFromImage(row,column)
saturation = getSaturationPixelFromImage(row,column)

value = geValuePixelFromImage(row,column)

2. avg = (hue+saturation+value)/3

3. if(avg<THRESHOLD)
paint white
else
paint black.

4. end [17]

A Brief revision of implemented algorithms used in gesture recognition: Gestures are detected by using CAMshift algorithm and background subtraction algorithms, All these necessary algorithms are provided by OPENCV and JAVACV. They are as follows:

3.Camshift Algorithm

The Continuously Adaptive Mean Shift Algorithm (CamShift) is an adaptation of the Mean Shift algorithm for object tracking that is intended as a step towards head and face tracking for a perceptual user interface. In this paper, we review the CamShift Algorithm and extend a default implementation to allow tracking in an arbitrary number and type of feature spaces.

In order to compute the new probability that a pixel value belongs to the target model, we weight the multidimensional histogram with a simple monotonically decreasing kernel profile prior to histogram back-paperion. [7]

The CamShift algorithm can be summarized in the following steps (Intel Corporation, 2001);

1. Set the region of interest (ROI) of the probability distribution image to the entire image.

2. Select an initial location of the Mean Shift search window. The selected location is the target distribution to be tracked.

3.Calculate a color probability distribution of the region centred at the Mean Shift search window.

4.Iterate Mean Shift algorithm to find the centroid of the probability image. Store the zeroth moment (distribution area) and centroid location.

5.For the following frame, center the search window at the mean location found in Step 4 and set the window size to a function of the zeroth moment. Go to Step 3. [7]

Other algorithms we have used in our paper are Background subtraction algorithm and Tracking objects in image algorithm which were directly used and defined in javacv and opencv libraries. [6]

4. Background Substraction Algorithm

Background subtraction (BS) is a common and wid ely used technique for generating a foreground mas k (namely, a binary image containing the pixelsbel onging to moving objects in the scene) by using sta tic cameras.As the name suggests, BS calculates th e foreground mask performing a subtraction betwe en the current frame and a background model, cont aining thestatic part of the scene or, more in genera l, everything that can be considered as background given the characteristics of the observed scene.[18]
Background modeling consists of two main steps: Background Initialization and Background Update. In the first step, an initial model of the background is computed, while in the second step that model is updated in order to adapt to possible changes inthe scene.Several algorithms were introduced for this p urpose. OpenCV has implemented three such algori thms that are Background Subtractor MOG, Background Subtractor MOG2, Background Subtractor GMG. Here we are using the first one Background Subtractor MOG.[18]
BackgroundSubtractorMOG Algorithm**:**
1.First, three Mat objects are allocated to store the current frame and two foreground masks,obtained by using two different BS algorithms.
2.Two cv::BackgroundSubtractor objects will be us ed
to generate the foreground masks. In this example,

default parameters are used, but it is also possible to declare specific parameters in the create function.[18]

3.The command line arguments are analysed. The u ser can chose between two options::video files (by choosing the option -vid);
image sequences (by choosing the option -img

4.Suppose you want to process a video file. The vid eo is read until the end is reached or the user presse s the button 'q' or the button 'ESC'.

5.Every frame is used both for calculating the foreg round mask and for updating the background. If yo u want to change the learning rate used for,updatin g the background model, it is possible to set a speci fic learning rate by passing a third parameter to the 'apply' method.[18]

6.The current frame number can be extracted from the cv::VideoCapture object and stamped in the top left corner of the current frame. A white rectangle is used to highlight the black colored frame number

7.We are ready to show the current input frame and the results.

The same operations listed above can be performed using a sequence of images as input. The processI mage function is called and, instead of using a cv::VideoCapture object, the images are read by us ing cv::imread , after individuating the correct path for the next frame to read.[18]

Besides these algorithms, the main concept which is proposed by us for conversion of gestures to text and speech is by using Free TTS Library.

5. Hand and Finger Detection

Appropriate library can be used as the basis of other shape analyzers, which I'll illustrate here by extending it to detect a hand and fingers. In Fig2, Person is wearing a black glove on my left hand. His Handy application attempts to find and label the thumb, index, middle, ring, and little finger. Yellow lines are drawn between the fingertips and the center-of-gravity (COG) of the hand.[5] ,[1]

Fig2. Detecting a Left Hand and Fingers.[1]

The demonstration of fig. determine suitable HSV ranges for the black glove. These ranges are loaded by Handy prior to executing the steps shown in Fig3 to obtain the hand's contour, its COG, and orientation relative to the horizontal. To do this we use following two function provided:

Use the function findContours
Use the function drawContours [1]

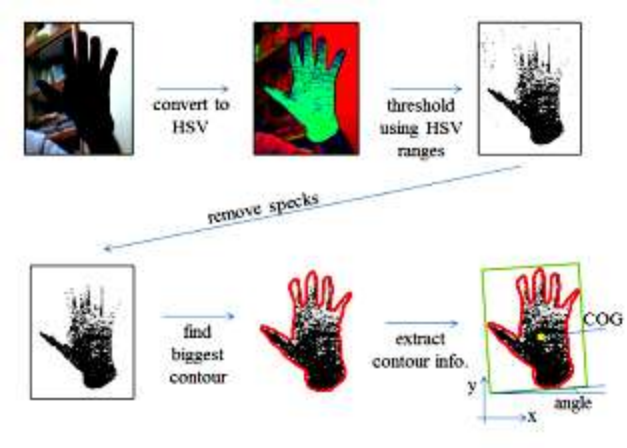

Fig3. Finding the Hand Contour.[1]

The various stages in this are almost identical to those carried out by the ColorRectDetector.findRect() method. However, Handy continues processing, employing a convex hull and convexity defects to locate and label the fingertips within the hand contour. These additional steps are shown in Fig4.

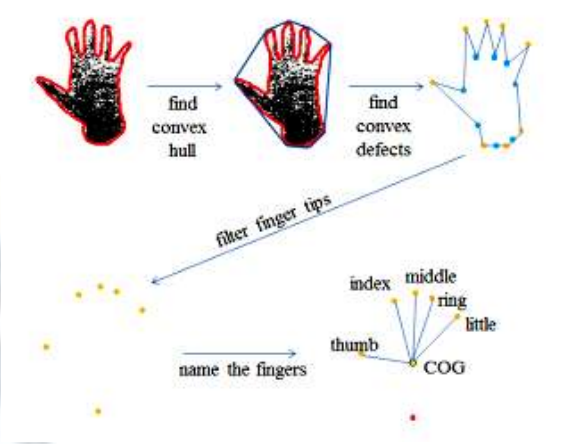

Fig4. Finding and Labeling Fingertips.[1]

The hull and defects are obtained from the contour with standard operations, which I'll explain below. However, the final step of naming the fingers utilizes a rather hacky strategy that assumes the contour's defects are for an out-stretched left hand. The thumb and index finger are located based on their angular position relative to the COG, and the other fingers are identified based on their position relative to those fingers. This process is rather fragile, and can easily become confused, as shown in following Fig5.[1]

Fig5. A Misidentified Middle Finger.[1]

6. Gesture Detection

The Handy application stops short at converting the named fingertips into gestures, which would require an analysis of how the fingers move through space over time.

Preliminary tests show that Handy can only identify gestures reliably when they involve an outstretched thumb and/or index finger, perhaps combined with other fingers. Gestures of this type include "victory", "wave", "good", "point", and "gun" shown in Fig6 .[5]

Fig6.Gestures Suitable for Handy-style Detection.[1]

7. Pattern Recognition

Pattern recognition is the field of engineering or sometimes classified as sub-section of machine learning with the goal of replicating human recognition and classification skills with the use of computer algorithms. One can observe that humans do pattern recognition with great ease, for example, colour coding, we can separate a set of coloured balls into different colour groups with no effort. Another fascinating example is our ability to recognize people seen only a few times or people seen a long time ago. In engineering fields like image processing, target identification and so on. the task of the pattern recognition engineer is to identify different sections of the image based on a certain property, for example, to classify a land image into a urban and rural areas. Recently, the efforts to automate these process have increased, especially, due to availability of computing power and advances in machine learning. Pattern recognition algorithms are usually known as classifiers. A few popular algorithms are k-Nearest Neighbour classifier, k-Means clustering.

The process consists of three major steps after data acquisition. Datasets for pattern recognition can be from a wide range of sources like satellite sensor data, ground based sensor data, medical images and so on. Once the dataset is acquired it is preprocessed, so that it is suitable for subsequent sub-processes. Next step is feature extraction, in which, the dataset is converted into a set of feature

vectors which are supposed to be representatives of the original data. These features are used in the classification step to segregate the data points into different classes based on the problem.[19]

Steps in pattern recognition-

1) Preprocessing

One of the most common preprocessing steps done in field of pattern recognition are normalization to zero mean and unit variance, especially for 1-D datasets. In the field of remote sensing most common preprocessing step required is re-gridding, which is basically assigning a spatio-temporally uniform grid to raw data. In many image processing applications, it is desirable to have a uniform spatial grid for the pattern recognition process. However, satellite datasets usually have non-uniform grid, this problem can be rectified by re-sampling the spatial data by either interpolation or averaging to an uniform grid. Another common method used is spatial interpolation, as most of the datasets acquired are usually full of missing data points. The problem of missing data points is well known in statistics and this problem can be overcome by using a slew of techniques from simple averaging to advanced spectral analysis methods.[19]

2) Feature Extraction

The main goal of feature extraction is to reduce the data dimensionality and properly represent the original data in feature space. Features useful for classification process can be simple features like RGB values in color images, or complex features like energies from the Fourier Transform or Wavelet Transform of a time series. The feature extraction process usually consists of three steps:

1) Feature construction is the step in which features are constructed from linear or non-linear combination of raw features.

2) feature selection process is done using techniques like relevancy ranking of individual features and

3) feature reduction process is used to reduce the no. of features especially when too many features are selected compared to the no. of feature vectors. These three steps are not mandatory in the feature extraction process.

a) Feature Construction
b) Feature Selection
c) Feature Reduction [19]

3) Classification

It evaluates the features presented and further makes the final decision

a) Training
b) Cross Validation
c) Testing [19]

8. Free TTS Concept

FreeTTS is a speech synthesis system written entirely in the Java™ programming language. It is based upon Flite: a small run-time speech synthesis engine developed at Carnegie Mellon University[8].

Text to speech conversion involves Text-to-speech (TT-S) systems based on the concatenation of speech units need a prosodic modification algorithm to adjust the prosodic features of the stored speech units to the desired output values.TTS application is not so easy as it requires more effort by the developer in cases text preprocessing where a text may be inputted in an ambiguous from by the different users, pronunciation problem where different words pronounced in different ways [9]. In this mostly homographs create much problem during pronunciation task. Lastly, the problem of prosody which includes the intonation, stress and duration which is the major challenging problem from many years.The text to speech conversion may be done in different steps: Text preprocessing, text analysis, text tokenization, prosody generation and then the speech synthesis using various algorithms [10].

The steps followed to convert text to speech are described in Fig7-

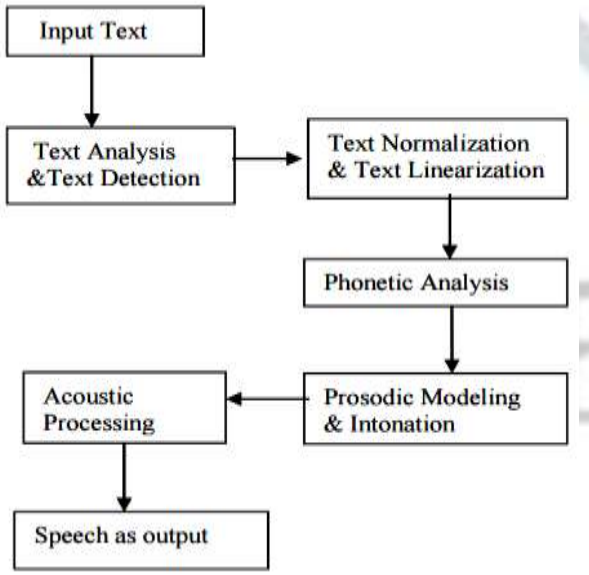

Fig7. Steps in TTS system [13] [14]

A. Text analysis and detection

The Text Analysis part is preprocessing part which analyse the input text and organize into manageable list of words. It consists of numbers, abbreviations, acronyms and idiomatics and transforms them into full text when needed. An important problem is encountered as soon as the character level : that of punctuation ambiguity (sentence end detection). It can be solved, to some extent, with elementary regular grammars

Text detection is localize [9] the text areas from any kind of printed documents. Most of the previous researches were concentrated on extracting text from video. We aim at developing a technique that work for all kind of documents like newspapers, books etc

B. Text normalization and linearization

Text Normalization is the transformation of text to pronounceable form. Text normalization is often performed before text is processed in some way, such as generating synthesized speech or automated language translation. The main objective of this process is to identify punctuation marks and pauses between words. Usually the text normalization process is done for converting all letters of lowercase or upper case, to remove punctuations, accent marks , stopwords or "too common words "and other diacritics from letters. The main 4 phases of Text Normalization are:

(i). Number converter: Number is pronounced differently in different situations.

(ii). Abbrevation converter: Abbreviations area changed to full textual format.

(iii). Acronym converter: Acronyms are replaced by single letter components.

(iv). Word segmentation: Sentences are a group of word segments. Special delimiter to separate segments. (i.e. „||"). Segments can be an acronym, a single word or a numeral. Punctuation marks are also identified .

Linearization is the process of giving a hyper text link to give the user a quick overview of the page. Then the TTS system will help to read out the linearized data.This feature helps in selecting the text and reading and also to list the links in the hyper text.

C. Phonetic analysis

Phonetic Analysis converts the orthographical symbols into phonological ones using a phonetic alphabet. Basically known as "grapheme-to-phoneme" conversion. Phone is a sound that has definite shape as a sound wave. Phone is the smallest sound unit. A collection of phones that constitute minimal distinctive phonetic units are called Phoneme. Number of phonemes is relatively smaller than the graphemes, only 44. Pronunciation of word based on its spelling has two approaches to do speech synthesis namely (a)Dictionary based approach (b) Rule based approach. A dictionary is kept were It stores all kinds of words with their correct pronunciation, it"s a matter of looking in to dictionary for each word for spelling out with correct pronounciation. This approach is very quick and accurate and the pronounciation quality will be better but the major drawback is that it needs a

large database to store all words and the system will stop if a word is not found in the dictionary.

The letter sounds for a word are blended together to form a pronunciation based on some rule. Here main advantage is that it requires no database and it works on any type of input. same way the complexity grows for irregular inputs [15]

D. Prosodic modelling and intonation

The concept of prosody is the combination of stress pattern , rhythm and intonation in a speech. The prosodic modeling describes the speakers emotion. Recent investigations suggest the identification of the vocal features which signal emotional content may help to create a very natural [10] synthesized speech. Intonation is simply a variation of speech while speaking. All languages use pitch, as intonation to convey an instance, to express happiness, to raise a question etc. Modelling of an intonation is an important task that affects intelligibility and naturalness of the speech. To receive high quality text to speech conversion, good model of intonation is needed. Generally intonations are distinguished as (i) Rising Intonation (when the pitch of the voice increases) (ii) Falling Intonation (when pitch of the voice decreases) (iii) Dipping Intonation (when the pitch of the voice falls and then rises) (iv) Peaking Intonation (when the pitch of the voice raises and then falls) [15]

E. Acoustic processing

The speech will be spoken according to the voice characteristics of a person, There are three type of Acoustic synthesing available

(i).Concatenative Synthesis (ii).Formant Synthesis (iii).Articulatory Synthesis

The concatenation of prerecorded human voice is called Concatenative synthesis, in this process a database is needed having all the prerecorded words .The natural sounding speech is the main advantage and the main drawback is the using and developing of large database.

Formant-synthesized speech can be constantly intelligible .It does not have any database of speech samples. So the speech is artificial and robotic.

Speech organs are called Articulators. In this articulatory synthesis techniques for synthesizing speech based on models of the human vocal tract are to be developed. It produces a complete synthetic output, typically based on mathematical models [15]

IV. APPLICATION

Our proposed software can be made in use at the institution for deaf and dumb people, were the deaf can understand the gesture converted text and at same time dumb can understand the gestures converted to voice.

V. CONCLUSION

This paper captures hand picture by using webcam or any available laptop and converts it into voice and text. For conversion into text to speech we have used FreeTTS library and for gesture detection, opencv and javacv are used in java.

Early, there were developed recognition systems but they had one limitation of background color, were the gestures were being captured. That is if background color and hand glove color is same then it is difficult to recognize and track hand but the proposed system have overcome this drawback by simply providing option to choosing color manually to user. This option appears every time the software is run for the first time, ie- after login of the user.

VI. FUTURE WORK

The proposed system can also be implemented for entire video shoot of some continuous gestures recognized and then converting these continuous gestures into corresponding sentences and also voice ie- speech.

REFERENCES

[1] https://fivedots.coe.psu.ac.th/~ad/jg/nui055/handy.pdf
[2] G. R. S. Murthy, R. S. Jadon. (2009). ―A Review of Vision Based Hand Gestures Recognition,‖ International Journal of Information Technology and Knowledge Management, vol. 2(2), pp. 405-410.
[3] P. Garg, N. Aggarwal and S. Sofat. (2009). ―Vision Based Hand Gesture Recognition,‖ World Academy of Science, Engineering and Technology, Vol. 49, pp. 972-977.
[4] http://docs.opencv.org/
[5] http://docs.opencv.org/modules/core/doc/intro.html
[6] B. DeRuyter and E. Pelgrim, ―Ambient Assisted-Living Research in Care- Lab,‖ Interactions, vol. XIV, no. 4, pp. 30–34, 2007.
[7] http://crpit.com/confpapers/CRPITV36Allen.pdf
[8] http://freetts.sourceforge.net/docs/index.php#what_is_freetts
[9] Eduardo R. Banga and Carmen Garcia-Mateo, Shape-invariant pitch- synchronous text to speech Conversion
[10] Suresh Kumar Thakur and K.J. Satao, Study of Various kinds of Speech Synthesizer Technologies and Expression for Expressive Text To Speech Conversion System.
[11] Michael H. O'Malley and Berkeley Speech Technologies, Text To Speech Conversion Technology.
[12] M. Nageshwara Rao, Samuel Thomas, T. Nagarajan, and Hema A. Murthy, Text-to-Speech Synthesis using syllable-like units.
[13]Yousif A.El-Imam and Karima Banat, Text to Speech conversion on Personal Computer
[14]Silvio Ferreia,Celina Thillou, Bernaud Gosselin, "From Picture to Speech: an Innovative Application for Embedded Environment"
[15]http://www.ijsce.org/attachments/File/v2i1/A0435022112.pdf
[16]http://www.tutorialspoint.com/java_dip/basic_thresholding.htm
[17]http://www.samundra.com.np/threshold-image-java-code/104
[18]http://docs.opencv.org/master/db/d5c/tutorial_py_bg_subtraction.html#gsc.tab=0
[19]https://en.wikiversity.org/wiki/Pattern_Recognition
[20]Digital_image_processing_2nd ed.pdf

AUTHOR'S BIBLOGRAPHY

Kajal B. Borole pursuing Bachelor of Engineering in Computer Engineering and will complete the degree of BE in this year 2016 from Government College of Engineering, Jalgaon.

Bhagyashree D. Patil pursuing Bachelor of Engineering in Computer Engineering and will complete the degree of BE in this year 2016 from Government College of Engineering, Jalgaon.

H D. Gadade received B.Tech in Computer Engineering in year 2004 from Dr. B A Tech. University, Lonere. Also received M.Tech in Computer in year 2008 from University of Pune. He is working as a Assistant Professor in Department of Computer Engineering in the Institute Government College of Engineering, Jalgaon.

